

# Aplicació multidispositiu per a la gestió de les pràctiques externes

David Hernando Jurado

**Resum**– En tots els estudis de grau s'han incorporat com a matèria les pràctiques externes. En aquest activitat hi intervenen molts actors diferents: empreses, tutors d'empresa, tutors acadèmics, estudiants, gestió acadèmica... i es fa indispensable disposar d'eines per a la seva gestió. Amb aquest projecte es vol facilitar la tasca del coordinador en aquesta gestió. GPE és una aplicació multi-dispositiu de gestió de les pràctiques externes del Grau en Enginyeria Informàtica de l'Escola d'Enginyeria desenvolupada amb les millors pràctiques de desenvolupament de software i últimes tecnologies web del mercat. Amb aquestes tecnologies s'ha pogut desenvolupar una aplicació robusta als canvis i fàcil de mantenir. A més a més, l'aplicació té un aspecte agradable alhora que funcional sense el cost d'un equip de dissenyadors.

**Paraules clau**– Angular, Laravel, PHP, MySQL, MVC, gestió, pràctiques en empresa, REST, JSON, Javascript, multi-dispositiu, responsive

**Abstract**– Every degree has added supervised externship as a subject in their curriculum. A variety of actors take part in this activity: companies, company tutors, academic tutors, students, academic management... and having a tool to manage every step becomes mandatory. Lowering the hassle of this task to the coordinator of the subject is the main goal in this project. GPE is a supervised externships manager multi-device application for the Engineering School at UAB. It is developed using best practices in software development and state of the art web technologies. Using these technologies it has been possible to develop an maintainable and robust to changes application. Furthermore, the application has a professional look and feel and strong functionality without the cost of a design team.

**Keywords**– Angular, Laravel, PHP, MySQL, MVC, management, supervised externships, REST, JSON, Javascript, multi-device, responsive

## 1 INTRODUCCIÓ

LES pràctiques curriculars en empreses externes són una activitat comuna a totes les universitats per a que els estudiants puguin sortir del grau amb experiència laboral.

En el cas de l'Escola d'Enginyeria, s'ofereix les pràctiques com una assignatura optativa amb un valor de 12 crèdits i 300h de durada [1]. Al grau d'Enginyeria Informàtica, aquest curs hi ha 100 estudiants matriculats que estan en procés de fer estades en empreses o organitzacions externes a la universitat.

El procés que es segueix a l'assignatura suposa fer di-

ferents passos. Primer de tot l'estudiant ha de buscar una empresa on realitzar l'estada de pràctiques curriculars. Per a facilitar aquest procés es publiquen ofertes provinents d'empreses interessades que envien propostes a l'Escola i que la coordinació considera adients. L'estudiant també pot buscar una empresa per ell mateix o trobar-la pel portal Nexus de Treball Campus. Si l'estudiant fa una proposta, ja vingui de Treball Campus o d'una empresa externa, aquesta ha de ser abans validada per la coordinació abans de començar-la. Per a poder-la validar s'ha de un document amb el Pla de Treball.

Quan l'estudiant ja té un lloc d'estada de pràctiques vàlid, se li assigna un tutor acadèmic que controlarà i donarà suport a l'estudiant durant l'estada.

El següent pas és la redacció i firma del conveni de pràctiques. En cas de que la proposta vingui de Treball Campus, l'estudiant ha de proporcionar al coordinador una còpia del conveni que el portal proporciona. En cas contrari és l'Escola qui elabora el conveni que l'estudiant i l'em-

---

E-mail de contacte: david.hernando@e-campus.uab.cat, dhj111291@gmail.com

Menció realitzada: Tecnologies de la Informació

Treball tutoritzat per: Jordi Pons Aróztegui (dEIC)

presa firmen. Amb el conveni entregat i firmat l'estudiant ja pot començar l'estada de pràctiques. Al llarg de l'estada el tutor acadèmic i l'estudiant poden trobar-se per fer un seguiment de la feina desenvolupada.

Quan arriba el final de l'estada de pràctiques l'estudiant ha de fer una memòria sobre les activitats que ha realitzat, tot detallant-les i valorant-les. També es precisa que el tutor de l'empresa té assignat l'estudiant entregui un informe de la seva valoració sobre les activitats que ha fet l'estudiant i de l'assoliment de les competències.

Quan s'han entregat els dos documents, el tutor acadèmic fa un informe final. Els dos informes es lliuren al responsable de l'assignatura que fa el càlcul de la nota final i l'entra a l'expedient. Per últim s'avis a l'estudiant que ha rebut la nota i se li demana que ompli una enquesta de valoració.

Com es pot veure, es tracta d'una activitat on hi ha molts passos a seguir i en el que intervenen força actors. És indispensable, doncs, que hi hagi una bona coordinació que asseguri que tot es fa correctament, que no es salti cap pas i en la mesura del possible, en el menor temps possible entre pas i pas.

## 2 SITUACIÓ ACTUAL

Actualment només hi ha un responsable de l'assignatura que s'encarrega de gestionar el procés complet de les estades per a tots els estudiants.

La informació de les propostes que arriben de les empreses s'escriuen en una plantilla predefinida en format Word i es passen a format pdf per a ser publicades al campus virtual com a mitjà de difusió.

Si un estudiant vol fer una proposta se li proporciona la mateixa plantilla, per a que l'ompli amb l'ajuda de l'empresa implicada. Un cop omplerta s'envia per mail al responsable, que s'encarrega d'informar a l'estudiant si la proposta és vàlida.

Quan la proposta és acceptada es procedeix a fer el conveni. Si es tracta d'una proposta de Treball Campus es demana una còpia del conveni proporcionat per ells firmat per totes les parts, sinó el conveni el fa l'Escola i se li envia a l'estudiant per a que el signin ell i la empresa implicada. El responsable de l'assignatura va guardant en un document Excel la informació més important de les diferents propostes, indicant l'estat en el que es troben. A més a més de guardar les dates d'inici i final i la informació de contacte dels tutors.

Uns dies abans d'acabar l'estada, des de la coordinació s'envia per mail un avís a l'estudiant per a que prepari la memòria (juntament amb una plantilla) i al tutor de l'empresa per a que empleni l'informe de l'estudiant (també amb la plantilla adjunta). Quan el responsable ha rebut els dos documents, envia un mail al tutor acadèmic corresponent amb la memòria de l'estudiant per a que faci la seva valoració.

Finalment quan acaba l'estada i el tutor acadèmic entrega el seu informe al responsable, aquest s'encarrega d'entrar a un altre document excel les notes i la resta de dades de l'informe, deixant com a últim pas l'entrada de la nota a l'expedient.

Es conclou doncs que hi ha molts passos en els que el coordinador ha d'intercanviar informació i arxius amb la resta d'actors, seguint un ordre i complint uns terminis. Actualment el control es fa a través d'arxius excel i de for-

ma manual. Veiem doncs la necessitat d'implementar una aplicació que doni suport al coordinador per a dur a terme aquestes tasques.

## 3 OBJECTIUS

Els objectius generals a assolir en el projecte són:

1. Facilitar la tasca de gestió i administració dels convenis de pràctiques en empresa, ja que el volum d'estudiants va creixent cada any, i amb la metodologia que hi ha ara és molt fàcil perdre el control de tots els procediments que hi ha implicats a l'hora de gestionar les pràctiques d'un estudiant.
2. Centralitzar la informació relacionada amb: empreses, propostes, convenis, estudiants, tutors acadèmics i tutors d'empresa.
3. Facilitar el seguiment, per part del coordinador, de tot el procés, mitjançant avisos automàtics quan s'està apunt d'acabar el temps estipulat d'una estada. Alhora facilitar la detecció de manca d'algun document relacionat amb l'estada.

Aquests, a nivell funcional es poden veure com:

- Gestionar les propostes.
  - Tenir en compte els orígens de les propostes, que poden venir d'empreses, d'estudiants, o de Treball Campus. Cada origen tindrà uns requisits diferents.
  - Visió de propostes que no han tingut estudiants assignats i han estat actives durant un determinat temps.
- Gestionar els convenis estudiant-empresa.
  - Enviament d'avisos a l'aplicació i per mail al responsable i a l'estudiant quan l'estudiant ha realitzat l'estada de pràctiques segons les hores establertes pel conveni i quan falta algun document per lliurar.
  - Gestionar quin és el tutor associat per a l'estada de l'estudiant i les seves dades. També poder enviar mails al tutor de l'estudiant relacionats amb els documents a lliurar pel tutor.
  - En tot moment poder veure quins documents han sigut lliurats o no per a cada estada.
  - Visió de les estades que han d'acabar en els propers dies.
  - Visió d'estudiants que encara no han sigut assignats.

## 4 ESTAT DE L'ART

Avui en dia hi ha molts gestors de contingut que presenten solucions per a problemes semblants al que busquem amb tecnologies web com Joomla[2], Wordpress[3] o Drupal[4]. El problema que tenen aquests gestors és que la seva personalització a nivell de lògica de negoci té certes limitacions

que no permetrien aconseguir satisfer el nivell de requeriments. A més a més, estan orientats a la generació de contingut per a pàgines públiques i no aplicacions web complexes, que és del que tracta el projecte.

És remotament possible que es poguessin arribar a satisfer els requeriments modificant el codi base dels gestors, fent que no es poguessin actualitzar en un futur ja que els nostres canvis podrien deixar de funcionar.

La plataforma Nexus de Treball Campus està orientada a la gestió de les pràctiques però només dona suport (informàticament) a la part de l'empresa i l'estudiant, concretament per la gestió de pràctiques extracurriculars, i el nostre projecte està pensat per a la gestió de les pràctiques curriculars on també hi ha la figura del tutor acadèmic i s'avalua a l'estudiant. A més a més, aquesta plataforma tampoc permet la figura del coordinador de pràctiques que pugui gestionar i validar les propostes dirigides als estudiants de la seva titulació. Com s'ha dit, està pensada per pràctiques no curriculars.

També existeixen borses de treball com Infojobs[5] que fan servir la mateixa lògica d'ofertes, però només utilitzen la lògica per apuntar-s'hi. La resta del procés es fa de forma externa. Algunes de les propostes pròpies d'estudiants provenen d'ofertes que s'han trobat al portal Infojobs.

En resum, el que es pot veure de les tecnologies existents és que no centralitzen la lògica de negoci per a tots els interessats de les pràctiques, ja que per la banda de l'Escola, es vol poder controlar l'estada de pràctiques des de l'oferta fins a l'avaluació, així com una revisió final del procés per a millorar les experiències dels tutors i estudiants.

Hi ha moltes universitats que han implementat algun software propi per a la gestió de pràctiques per a alumnes i empreses. Un exemple és la universitat Rey Juan Carlos, on s'hi pot consultar els manuals d'usuari tant per a estudiant, com per a empreses [6].

S'ha pogut veure que la gestió de les ofertes que passen per diferents estats en els que intervenen un conjunt d'actors es podria modelar com a un BPM (Business Process Management) [7]. Tot i això es considera que no és necessari l'ús d'un SW de BPM per a assolir els objectius del projecte.

## 5 METODOLOGIA

Per al desenvolupament del projecte s'ha intentat seguir un model de desenvolupament Agile, tot i que no s'ha pogut seguir estrictament. Aquest model de desenvolupament es basa en el model iteratiu, però amb algunes diferències.

En primer lloc en el model Agile es busca una forta implicació del client que utilitzarà finalment l'aplicació.

En segon lloc busca allunyar-se del procés estricte i tediós dels models antics com el de cascada o iteratiu clàssic, en els que s'ha de documentar exhaustivament cada pas en el desenvolupament del producte. Sinó que s'interactua amb el client després de desenvolupar cada funcionalitat, es rep feedback del client i es fan les possibles modificacions en una nova iteració sense tenir que refer tot el que s'ha desenvolupat.

En equips de desenvolupadors a més a més de seguir el model Agile es fan servir mètodes específics com Scrum[8] o Extreme Programming[9], que donen directrius a l'hora d'estructurar les tasques i reunions que es fan durant el desenvolupament.

En el cas d'aquest projecte no s'ha seguit un mètode específic, sinó que s'ha desenvolupat la aplicació intentant seguir aquesta filosofia, ja que només ha participat una persona en el desenvolupament.

En la figura 1 es pot veure un exemple gràfic de la filosofia d'aquest mètode.

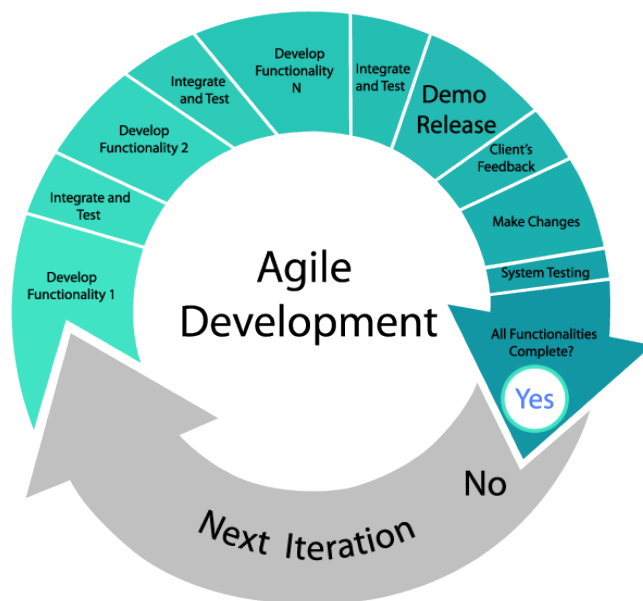


Figura 1: Diagrama del model de desenvolupament Agile[10]

S'ha utilitzat l'eina Trello [11] per a l'organització de les tasques del projecte, de tracking de bugs que s'han pogut trobar i tracking de l'estat del projecte, que permeten al client veure com es troba el projecte de forma visual. S'han organitzat un seguit de llistes per al seguiment del desenvolupament:

**TODO:** tasques a desenvolupar.

**IN PROGRESS:** tasques en les que s'hi està dedicant l'esforç actualment.

**VERIFICATION:** tasques realitzades que necessiten ser provades per a ser pujades.

**DONE NOT UPLOADED:** tasques desenvolupades i provades localment però que no són accessibles al client.

**BUGS:** errors i comportaments inesperats de l'aplicació.

**DONE:** tasques realitzades i visibles al client i que pot provar lliurement.

En la figura 2 es pot veure una captura on es pot apreciar l'estructuració mencionada i la facilitat que dona l'aplicació, ja que tots els elements són arrossegables mitjançant dragAndDrop i s'hi poden associar imatges i checklists.

### 5.1 Requisites

#### 5.1.1 No Funcionals

Els principals requisits no funcionals són que l'aplicació ha de ser compatible amb les tecnologies que tenen disponibles al servidor de l'Escola d'Enginyeria. Específicament són un

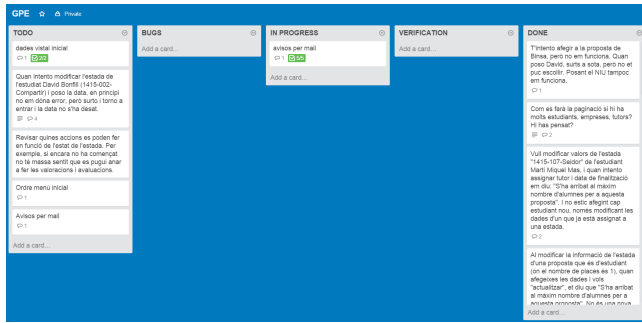


Figura 2: Captura del sistema de tracking de tasques del projecte, Trello

sistema de base de dades MySQL i la tecnologia de servidor PHP. També un requisit vital i central en el desenvolupament del projecte és que l'aplicació sigui multi-dispositiu.

### 5.1.2 Funcionals

El conceptes principals en els que es basen les pràctiques externes són les **propostes** i els **documents** generats a partir de l'estada d'un alumne en una empresa. A partir d'aquí els requisits s'han dividit en diferents apartats, en funció dels objectius que es volen aconseguir amb el projecte.

- **Propostes:**

En la Figura 3 es pot veure en forma de mapa conceptual el conjunt de requisits simplificats en relació al cicle d'estats d'una estada de pràctiques a una empresa.

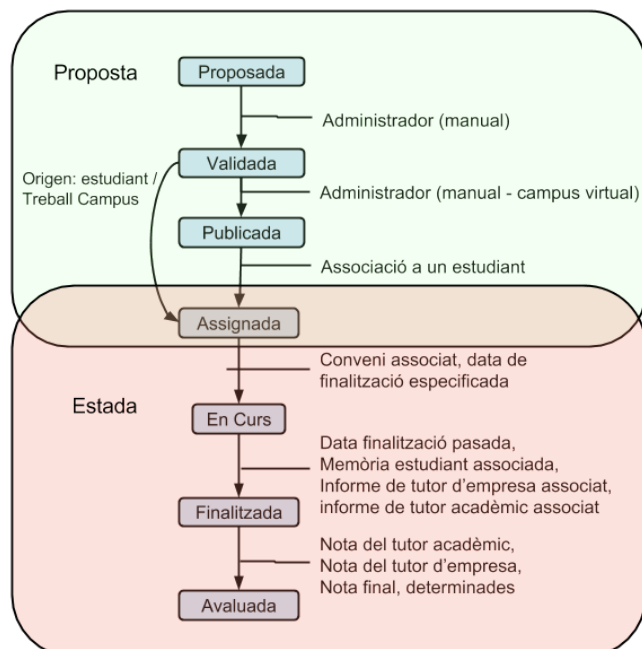


Figura 3: Diagrama d'estats d'una proposta - estada

Totes les propostes han de ser validades manualment per l'administrador de l'assignatura i se'ls hi ha d'associar el document d'oferta.

Un cop validada, el comportament d'una proposta pot variar segons el seu origen. Per una banda, si la proposta prové d'una empresa, aquesta pot ser publicada

i després podrà ser assignada a un estudiant. Per altra banda, si la proposta ve d'un estudiant, ja sigui de Treball Campus o externa, aquesta passa directament a ser assignada sense publicació.

A partir del punt en que una proposta ha sigut assignada a un estudiant, aquest pot començar l'estada. En el moment en que tingui el conveni, que pot ser l'entregat per gestió acadèmica o el proporcionat per Treball Campus. On aquest punt estaran ja establerts el tutor acadèmic, el tutor de l'empresa i la data de finalització.

Quan l'estada està en curs i es vol finalitzar, és obligatori que hagi passat la data de finalització especificada i s'hagi fet entrega de la memòria per part de l'estudiant, l'informe del tutor d'empresa sobre l'estudiant i l'informe del tutor acadèmic.

Un cop finalitzada l'estada, quedarà com a avaluada quan s'hagin especificat la nota final, la nota del tutor d'empresa i la nota del tutor acadèmic.

És possible una proposta tingui assignada més d'un estudiant, però el seguiment de l'estada per a cada estudiant ha de ser únic.

- **Documentació:**

S'ha de poder gestionar tota la documentació relacionada amb cada estada. Els principals documents, que ja s'han vist són:

- Conveni
- Memòria de l'estudiant
- Informe del tutor d'empresa
- Informe del tutor acadèmic

- **Avisos:**

S'ha de notificar al coordinador dels següents events: Enviar un recordatori si falta una setmana per a que una estada arribi a la data de finalització i no s'ha entregat l'informe del tutor d'empresa, o la memòria de l'estudiant. Enviar un recordatori en el moment en que s'ha fet l'entrega de l'informe del tutor d'empresa i de la memòria de l'estudiant per demanar l'informe al tutor acadèmic. De forma opcional enviament de mails als actors implicats de els lliuraments.

- **Valoracions de l'alumne:**

Es vol emmagatzemar les valoracions que fan els alumnes de les assignatures relacionades amb l'estada que han realitzat.

## 5.2 Tecnologies escollides

En aquesta secció s'explicarà quines tecnologies s'han escollit tant per al desenvolupament del servidor tant com el del client per tal de complir els requisits del projecte i els seus objectius.

En la figura 4 es pot veure visualment les tecnologies escollides.

### 5.2.1 Back-end (servidor)

Pel requisit no funcional de que el projecte sigui compatible amb els servidors de l'escola el sistema de base de dades serà MySQL.

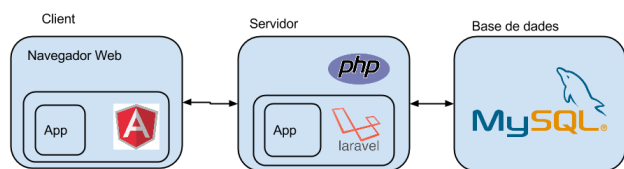


Figura 4: Diagrama de les tecnologies utilitzades en el projecte

Per al desenvolupament del servidor s'ha escollit el llenguatge PHP. A més a més, per a una aplicació de gestió és essencial estructurar el codi de forma que el seu manteniment sigui poc costós. Per tant, s'ha escollit el framework Laravel [12], un framework d'última generació que segueix les 'Best Practices' [13] del desenvolupament web i està basat en el patró Model-View-Controller (MVC).

### 5.2.2 Front-end (client)

El desenvolupament de l'aplicació es farà per a navegadors Web per tal de complir que l'aplicació sigui multi-dispositiu desenvolupant només una vegada. En la part del client (navegador web) s'ha escollit la tecnologia AngularJS [14], un framework JavaScript desenvolupat per Google que permet l'estructuració MVVM (Model-View-ViewModel) del codi en el client, això permet una separació de la presentació i la lògica de negoci clara, donant la possibilitat de construir codi reutilitzable. A més a més s'ha escollit un subconjunt del framework AngularJS per a l'aspecte de l'aplicació, Angular-material. Aquest subconjunt dona un aspecte uniforme i professional a l'aplicació i permet al desenvolupador estalviar temps en crear una UI agradable per l'usuari i ens dona components molt útils que permeten una interacció fàcil de l'usuari.

Angular-material compleix amb els estàndards Material[15] de Google per mantenir uniformitat entre aplicacions web i natives de mòbil.

A més a més, ens proporciona un sistema de reestructuració dels elements de les vistes en forma responsive[16][17], això ens permetrà desenvolupar la interfície d'usuari només una vegada definint les transformacions dels components en funció del dispositiu que carrega la pàgina.

### 5.2.3 Control de versions

En el desenvolupament de software actual es és indispensable un programa de control de versions per a tenir un tracking de quines versions del codi s'utilitza en l'entorn de producció i desenvolupament i control de creació de noves funcionalitats que podrien trencar el programa en un futur.

La seva efectivitat és màxima quan es treballa en un equip de múltiples desenvolupadors, però de totes formes és una eina molt útil per a projectes on només hi participa un únic desenvolupador.

Per a aquest projecte s'ha escollit Git[18] com a sistema de control de versions per la seva flexibilitat i control de l'històric de canvis en el codi i perquè el desenvolupador ja té experiència prèvia amb l'eina.

No s'ha seguit un model de branching concret, sent un únic desenvolupador el que interactua amb el projecte.

## 5.3 Disseny

Un cop es van captar els requisits inicials de l'aplicació s'ha dissenyat la base de dades per tal de modelar les dades de l'aplicació amb l'objectiu de satisfer els requisits. Als annexos es poden veure el diagrama entitat relació i lògic de la base de dades dissenyada.

Després es va passar a dissenyar l'aspecte visual de l'aplicació. Sent una aplicació de gestió aquest apartat no ha sigut molt prioritari. S'ha dissenyat una forma comú de presentar les dades i un menú per a l'accés de cada mòdul de l'aplicació. Després segons requisits s'han modificat les pantalles segons les dades requerides i casos d'ús implicats.

La pantalla que ha requerit més esforç a l'hora de dissenyar és la d'emplenament de dades de les estades, que inclou selecció de proposta, tutors, pujada de documents i avaluacions de l'estudiant de l'estada. Com que les estades estan directament relacionades amb els estudiants també es pot interactuar amb les dades de l'estudiant.

## 5.4 Desenvolupament

S'ha escollit una arquitectura de Web Services [19] amb RESTful [20], en el que servidor actua com una entitat que retorna dades amb les que l'aplicació client processa i interactua. Per a seguir aquest patró es desenvoluparà una Single Page Application (SPA) [21].

Aquesta arquitectura permet desacoblar la tecnologia del servidor dels dispositius clients, cosa que dona robustesa als canvis en el futur tant en el codi del client tant com del servidor, ja que si canviés la tecnologia del servidor només s'hauria de desenvolupar el nou servidor de forma que complís la interfície RESTful establerta i la lògica amb la que es va fer, no s'hauria de fer cap canvi en el client. I en el cas dels clients, qualsevol tipus de client pot consumir les dades del servidor sempre que segueixi les crides a l'estructura REST i pugui tractar el tipus de dades que es retornen, això inclou peticions ajax, curl, dispositius mòbils i altres servidors.

### 5.4.1 Back-end

Com s'ha especificat, el desenvolupament del servidor s'ha realitzat amb el framework MVC Laravel. Com que el servidor s'ha de desenvolupar en arquitectura RESTful, la part de la vista del patró MVC només s'utilitza per la vista de login i la de la pàgina principal sota la que s'executarà la SPA.

#### Injecció de dependències[22]

Per a la resta sí que s'ha seguit una arquitectura de model/controller. A més a més s'ha seguit el patró de injecció de dependències/inversió de control amb l'injector que proporciona Laravel. Això ens permet definir les dependències de les classes en els seus constructors mitjançant interfícies i indicant a la classe encarregada d'injectar les dependències que ha d'instanciar. És a dir deleguem la instanciació de les classes fora de la nostra aplicació, fent-la més fàcil de provar i facilitant la detecció d'errors. S'ha fet ús de la injecció de dependències per la creació dels serveis en els controladors que contenen la lògica de negoci, com la creació d'estades i ofertes, canvis d'estats en les estades, etc.

L'objectiu de programar d'aquesta forma és per millorar la testabilitat del codi, fer-lo més robust als canvis, ja que el codi que faci servir les interfícies dels serveis no ha de canviar encara que canviï el codi dels serveis, i veure fàcilment quines dependències tenen les classes.

### Autenticació

S'ha utilitzat el sistema de login que proporciona Laravel, que funciona amb correu electrònic i proporciona les millors pràctiques a l'hora de desar les contrasenyes, fent hashing amb l'algoritme bcrypt[23] i un salt generat aleatòriament. Un cop l'usuari s'ha logat s'utilitza la cookie amb la sessió per a la autenticació de la utilització de les crides REST. Aquesta pràctica no és ideal per a l'arquitectura RESTful, però s'ha decidit seguir així per evitar complicacions i allargar el temps del projecte.

### Estructura de crides REST

L'estructura de la majoria de crides HTTP al servidor segueixen el patró determinat a la taula 1:

Taula 1: Taula d'esquema de crides REST al servidor

Rebre Llistat	GET	/api/<entitat>
Rebre única entitat	GET	/api/<entitat>/<id>
Crear nova entitat	POST	/api/<entitat>
Actualitzar entitat	PUT	/api/<entitat>/<id>
Esborrar entitat	DELETE	/api/<entitat>/<id>

Les entitats amb cerques complexes segueixen el patró:  
GET /api/<entitat>/search?<parametre1>=<valor1>

&<parametre2>=<valor2>

En aquesta crida s'han definit uns paràmetres fixes per tal de controlar la paginació i l'ordenament. Page especifica la pàgina, orderby el camp a ordenar i dir la direcció de l'ordenament. Això, juntament amb un servei que s'encarrega de la paginació i ordenació, ha fet que totes les tasques de filtrat i ordenament siguin ràpides de desenvolupar. També s'ha seguit aquest esquema per a la càrrega d'entitats relacionades, per exemple,

GET /api/stay/<id>/document

ens retornaria totes les dades dels documents relacionats amb una estada, o

GET /api/stay/<id>/document/<idDoc>

ens retornaria les dades del document d'una estada.

El format del contingut de les crides s'ha fet tant com per les peticions tant com per les respostes en JSON [24], per tant els clients que interactuïn amb el servidor hauran de seguir aquest format. Per a la gestió d'aviso s'ha utilitzat un cron de unix [25] que cada minut crida a un script php en el que un scheduler que proporciona Laravel ens permet executar tasques recurrents de forma fàcil especificant-ne la hora i els intervals d'execució i aprofitant l'injector de dependències per a re-aprofitar el codi utilitzat en altres parts de l'aplicació. Per a l'aplicació s'ha establert que s'executi cada dia el procés de notificacions a les 5 del matí, de manera que cada dia quan l'usuari obri l'aplicació cada dia els veurà.

### Pujada d'arxius

S'ha desenvolupat un sistema de pujada d'arxius per a

les propostes i les estades que renombra els arxius i els estructura en un sistema de carpetes adient i lògic. El renombrament es fa de la següent forma:

<prefix>\_<data-pujada>\_<versio>.<extensio>

Fent el renombrament s'assegura consistència en els noms dels arxius, són més fàcils de trobar en cas de que s'hagin de buscar manualment i reforça la seguretat del servidor. La versió s'utilitza per tenir-ne una còpia en cas de que s'esborri un document inesperadament.

L'estructura d'arxius per les propostes es basa en establir en el prefix de l'arxiu el codi de la proposta i s'agrupen en una sola carpeta. En el cas de les estades les carpetes s'agrupen en el curs acadèmic en el que ha començat l'estada i en el següent nivell pel NIU de l'alumne. D'aquesta forma tenim agrupats els documents de les estades segons alumne d'una forma ràpida de trobar. Com a prefix s'ha utilitzat l'identificador únic generat per a l'estada.

### 5.4.2 Front-end

L'estructura que dona Angular és molt semblant a la dels frameworks de back-end, com Laravel. Una de les diferències que té és que no utilitza estrictament el concepte de Model. El que fa principalment una aplicació Angular és fer crides a un servidor i aquest retorna les dades a processar i enviar a la vista.

El concepte d'injecció de dependències també apareix en Angular, es pot aplicar exactament els mateixos conceptes que al servidor. D'aquí la coherència de triar un framework avançat al servidor, ofereix paritat en els conceptes aplicats a l'hora d'escriure el codi i dona lloc a un manteniment més fàcil.

L'estructura de l'aplicació s'ha organitzat en mòduls separables, on cada mòdul té les seves vistes, controladors i serveis (igual que en el servidor, són els que contenen la lògica del programa), i cada mòdul pot tenir dependències sobre altres mòduls. Per exemple, el mòdul d'estades té dependències sobre el mòdul de les propostes, el mòdul d'estudiants, el de tutors acadèmics i el de tutors d'empresa. Si un dels mòduls dels que depèn es desacobla de l'aplicació aquest mòdul deixaria de funcionar. Alhora els mòduls sense dependències es poden treure sense conseqüències en l'aplicació. Aquesta arquitectura de l'aplicació permet afegir noves funcionalitats de forma fàcil, un fet essencial perquè l'objectiu del projecte comprèn una petita part de tot el sistema que pot arribar a ser la gestió de pràctiques externes.

En la següent figura es pot veure un mapa de les dependències dels mòduls de l'aplicació.

Tota l'aplicació s'ha fet de forma que s'adapti tant a pantalles d'ordinadors d'escriptori com pantalles de dispositius mòbils. Angular-material proporciona una sintaxi mitjançant atributs HTML per amagar, redimensionar o canviar la forma en que es distribueixen els elements segons l'amplada del dispositiu. El menú principal també s'amaga i es mostra un botó que el desplega quan es visualitza l'aplicació des d'un dispositiu mòbil, pel que no es perd la navegació que es té en escriptori. En les figures 6 - 7 es pot veure de quina forma s'adapten els elements en escriptori i en dispositiu mòbil respectivament.

Per últim s'ha fet un ús ampli d'un concepte d'Angular anomenat directives. Consisteixen en encapsular en un element HTML, lògica i una plantilla HTML que poden ser



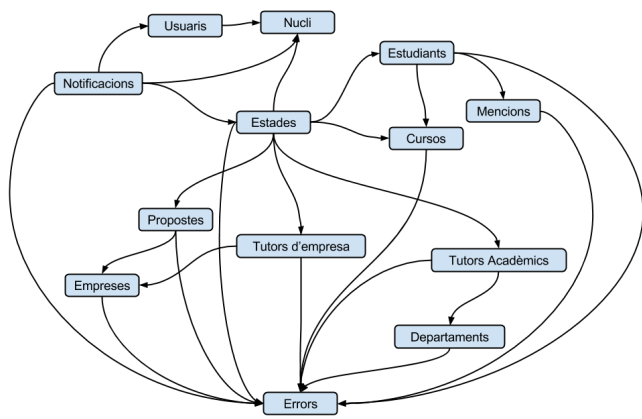


Figura 5: Diagrama de les dependències dels mòduls de l'aplicació

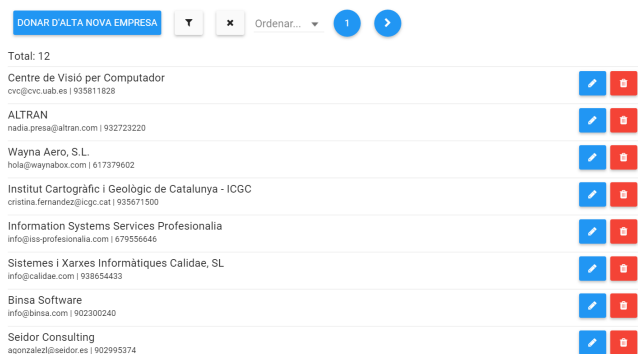


Figura 6: Captura de l'aplicació en escriptori



Figura 7: Captura de l'aplicació en dispositiu mòbil

reutilitzats al llarg de l'aplicació. Bàsicament seria una eina que proporciona una interfície per a donar re-usabilitat al codi. A més a més proporciona robustesa als canvis, ja que si s'hagués de canviar alguna part de dintre d'una directive aquest canvi es veuria reflectit a tota l'aplicació, pel que s'eliminen potencials bugs que puguin sorgir en codi duplicat.

### 5.4.3 Problemes trobats

Hi han hagut uns quants problemes que han allargat el desenvolupament de l'aplicació però que s'han pogut solucionar de forma correcta i sense la necessitat de crear 'hacks' o codi difícil d'entendre o potencial a generar bugs.

El principal problema ha sigut la serialització de dades a JSON a partir de les dades que es reben des de la base de dades. El driver MySQL que porta PHP per defecte retorna totes les dades de la BBDD en format de cadenes de caràcters. Però el client espera les dades formatades correctament, per exemple els camps que haurien de ser nombres sense cometes es retornen amb cometes, incomplint l'especificació de JSON[24], com que Angular espera dades que compleixen l'especificació, l'aplicació es comportava de forma inesperada. Això s'ha solucionat gràcies als castings que proporciona Laravel que permeten especificar el format en el que s'han de serialitzar els camps a JSON.

El segon problema que ha sorgit és l'assignació de dades, com la data de finalització d'una estada o de publicació d'una proposta. El navegador serialitza per defecte les dades en format UTC sense especificar la diferència horària, cosa que feia que les dates es desessin en la base de dades amb un dia enrere (l'aplicació espera dates en GMT+02). Aquest problema s'ha solucionat parsejant les dates que no compleixen el format esperat en el servidor i desant-les en el seu format correcte.

Un tercer problema bastant important ha sorgit en la pujada d'arxius amb dades associades, com els documents de les propostes en els que s'hi associava les dades de la proposta per a fer la pujada d'un sol cop passava un problema en el que també totes les dades es serialitzaven com a cadenes de caràcters i per exemple valors com null es representaven com 'null', o valors sense definir de javascript(undefined) es serialitzaven en comptes de ser ignorats. Això provocava errors greus a l'hora de desar les dades de les propostes i s'ha hagut de crear un servei que netejava les dades abans de desar-les a la base de dades.

Per últim, hi ha hagut un problema que ha portat més temps arreglar. El lloc web assignat per a l'aplicació en el servidor de l'escola estava sota la subruta "/dhernando", quan es va desenvolupar l'aplicació no es va tenir en compte que seria sobre una subruta i gran part de l'aplicació va deixar de funcionar. Això s'ha arreglat definint el tag <base> en la declaració del document principal HTML i migrant tot l'enrutament per part del client de forma relativa en comptes d'absoluta, que és com es va pensar en un principi. Per aplicar aquest canvi es va tenir que canviar totes les referències al lloc web dins del codi eliminant-hi el caràcter '/' per a que utilitzessin l'adreçament relatiu. Per sort, la utilització de serveis en Angular i la bona estructuració inicial del projecte van permetre tornar l'aplicació a la normalitat sense conseqüències i mantenint la paritat entre l'entorn de desenvolupament i l'entorn de producció.

## 5.5 Aplicació

Cada mòdul en el que s'ha dividit l'aplicació té la seva forma de presentar les dades en els llistats però seguint una base per al filtratge i paginació. A més a més les dades es poden editar i eliminar. La eliminació es un procés delicat, no es permet l'eliminació de dades que tenen dependències

en altres dades (per exemple no es pot eliminar una empresa que té estades en curs).

### 5.5.1 Propostes

El principal fet a destacar de les propostes és que es requereix la pujada del document associat. També hi ha la possibilitat de canviar-lo quan la proposta ja s'ha creat i el document s'ha actualitzat o ha canviat.

Figura 8: Pantalla de modificació de les dades de propostes

### 5.5.2 Estades

A la figura 9 es pot veure com seria el llistat d'estades, una de les pantalles més importants de l'aplicació. En el llistat

ID	Cod. prop.	Altrun	Data Fi	En Curs	Assignada	Avaluada
ID: 1	Cod. prop. 1516-004	Cristina Díez Marti	Data Fi: en 3 mesos (10/10/2015)	En Curs	Assignada	Avaluada
ID: 2	Cod. prop. 1516-005	Victor Asensio Casas	Data Fi: en 3 mesos (30/09/2015)	En Curs	Assignada	Avaluada
ID: 3	Cod. prop. 1415-107	Marti Miquel Mas Susi	Data Fi: fa 4 dies (23/06/2015)	En Curs	Assignada	Avaluada
ID: 4	Cod. prop. 1415-104	Xavier Oliván Crego	Data Fi: Sense determinar ()	En Curs	Assignada	Avaluada
ID: 5	Cod. prop. 1415-002	David Benfil Miralles	Data Fi: fa 5 dies (24/06/2015)	En Curs	Assignada	Avaluada
ID: 6	Cod. prop. 1415-117	Javier Gómez Domínguez	Data Fi: Sense determinar ()	En Curs	Assignada	Avaluada
ID: 7	Cod. prop. 1415-008	David Hernandez Jurado	Data Fi: Sense determinar ()	En Curs	Assignada	Avaluada

Figura 9: Captura del llistat d'estades amb visualització de documents entregats

tat es marca quins documents han sigut entregats de forma visual, l'estudiant de l'estada, la seva data de finalització i informació de la proposta que ha generat l'estada. També hi ha el sistema de filtrat i ordenació que s'ha esmentat a l'apartat de metodologia. L'usuari té un alt nivell de personalització a l'hora de veure les dades.

Existeixen dues formes d'assignar una estada, en el moment en que es crea una proposta que té d'origen Treball Campus o externa o quan s'assigna una proposta a un estudiant a partir de la pantalla principal d'estada.

Quan una estada s'ha creat el seu estat és assignada. Es pot passar a En Curs quan es compleixen els requisits. En cas de que no es compleixin es retorna un error especificant què es necessita. Aquest comportament es repeteix en els diferents estats pels que passa una estada.

Com s'ha esmentat en el disseny, la vista inclou les pestanyes de documentació, avaluacions i valoracions de l'estudiant de les assignatures relacionades amb l'estada. A la Figura 10 es pot veure com ha quedat la pantalla en format escriptori.

Figura 10: Pantalla de la gestió de les dades d'una estada

## 6 EXPOSICIÓ DELS RESULTATS

El resultat del projecte és una aplicació extensible i robusta als canvis en la que es poden gestionar les estades dels alumnes de forma visual i permet cerques que faciliten el treball del coordinador de l'assignatura.

També s'ha aconseguit un aspecte professional gràcies a les eines utilitzades sense una gran inversió de temps en aquest camp. L'aplicació és accessible des de qualsevol dispositiu que tingui un navegador web que pugui interpretar Javascript i a, més a més, la vista s'adapta segons el dispositiu.

La forma en la que s'ha desenvolupat l'aplicació també permet la creació de noves aplicacions, de forma ràpida ja que s'ha realitzat de tal manera que molts elements són reutilitzables.

Gràcies a l'aplicació de la metodologia Agile, el client ha interactuat amb l'aplicació des de les fases inicials i s'ha rebut feedback de coses que faltaven i d'errors que s'ha trobat. Això ens assegura que el producte resultant és satisfactori, i que cobreix les necessitats del client. Creiem que amb aquesta aplicació es facilita en gran mesura les tasques del dia a dia del coordinador de pràctiques.

## 7 CONCLUSIONS

El desenvolupament de l'aplicació ha anat bé en el que respecta al disseny de la lògica i la base de dades, ja que els requisits estaven clars des del principi i s'han definit bé. S'ha fet un bon disseny inicial però s'ha pres alguna decisió incorrecta en la forma d'implementar l'aplicació, el que ha provocat que la planificació inicial hagi hagut de ser reestructurada diverses vegades.

La planificació del projecte podria haver anat millor si s'hagués desenvolupat l'aplicació de forma "vertical". Amb això vull dir que en el desenvolupament es va fer primer tota l'estructura del servidor per després poder desenvolupar la part del client. Això va fer que el client no pogués interactuar amb l'aplicació des del primer moment i no veiés



els resultats fins que una part de l'aplicació era estable. Per exemple es podria haver desenvolupat primer el mòdul d'estudiants complet, provar-lo i veure'l amb el client per rebre'n feedback i poder anar avançant mòdul a mòdul de forma completa. Per sort, els resultats eren satisfactoris i s'ha pogut continuar i finalitzar el projecte sense problemes.

El que es pot concloure del desenvolupament de l'aplicació és que les noves tecnologies del desenvolupament web permeten construir servidors web que no estan destinats només a proporcionar html a navegadors web, sinó que proporcionen dades en formats estàndard que qualsevol dispositiu pot interpretar sense problemes.

Gràcies a això, si el projecte s'hagués de migrar a una aplicació nativa de dispositiu mòbil les implicacions serien mínimes per al servidor.

Per una banda el desenvolupament ha sigut una mica difícil al començament del projecte degut a la corba d'aprenentatge que ha suposat aprendre el funcionament de Laravel, però ha valgut la pena perquè un cop dominats els conceptes principals d'injecció de dependències i el patró Model-View-Controller, el desenvolupament de noves funcionalitats ha sigut molt ràpid. Es podria dir el mateix amb Angular però ja en tenia experiència anterior pel que no ha suposat problema ni cost addicional en temps d'aprenentatge.

Per altra banda, en aquest projecte he après la importància d'una bona estructuració en una aplicació de gestió, i de com ha evolucionat el desenvolupament d'aplicacions de gestió, passant de funcionar només en escriptoris tradicionals, a funcionar en qualsevol dispositiu comunicant-se amb servidors, afegint molts més conceptes i complicacions que s'han intentat alleugerar establint estàndards. També he après la importància de seguir un model de desenvolupament de software en el que el client tingui un paper actiu i el treball que això estalvia a l'hora de detecció i correcció d'errors ràpida, així com la correcta especificació dels requisits. Ja que quant més temps passa entre el desenvolupament d'una característica i la detecció dels seus errors més difícil és arreglar-los efectiva i ràpidament.

Com en qualsevol aplicació, hi ha molts aspectes que es poden millorar i, al tractar-se d'un problema complex, s'hi poden afegir nous mòduls. A continuació s'exposa algunes de les possibles millores i ampliacions que es poden fer.

## 7.1 Línies de millora

En primer lloc es podria internacionalitzar l'aplicació. Tot i que s'ha desenvolupat en català, si en el futur es volgués expandir el seu ús s'han deixat les facilitats dels frameworks utilitzats.

Una altra línia de millora seria la reestructuració dels menús d'una forma una mica més intuitiva per a facilitar la navegació de l'usuari.

Una millora molt útil seria mantenir una memòria dels filtres que s'apliquen als llistats, per a poder recuperar quan es tanca l'aplicació.

Per últim també es podria refer el login per a que utilitzés el NIU. O que utilitzés el sistema de login institucional de la UAB i, així no haver de recordar un usuari i contrasenya específics per a l'aplicació.

## 7.2 Línies d'amplicació

Es podrien fer mòduls de càlcul d'estadístiques i generació de reports de les notes dels estudiants, nombre de estades per empresa, estades per curs, etc.

Aquest projecte només abarca una part del marc de desenvolupament de les pràctiques externes.

El que s'ha implementat gira en torn a l'administració privada d'aquestes. Per tant, es poden proposar moltes ampliacions que resolguin altres aspectes de la gestió i desenvolupament de tot el procés.

L'aplicació està orientada per a que en el futur es pugui ampliar l'accés a estudiants per a buscar ofertes publicades i apuntar-s'hi. Que les empreses en puguin proposar sense tenir que fer el pas d'enviament per mail a l'administrador i que puguin contactar amb els estudiants directament; que els tutors puguin omplir els informes de valoració en l'aplicació; generar automàticament els convenis, etc.

El potencial de l'aplicació i de les dades que genera és molt gran. Fins i tot es podrien fer estudis de les valoracions de les empreses sobre els alumnes i de la contractació que fan per a poder arribar a futurs acords amb elles.

## AGRAÏMENTS

M'agradaria agrair al meu tutor del treball, Jordi Pons, per guiar-me durant el projecte, ajudar-me a millorar els meus informes i participar en el desenvolupament del projecte.

També vull agrair als meus pares el suport que m'han donat durant el procés i als meus amics per entendre que hi hagués alguns caps de setmana no podia sortir amb ells per acabar el projecte.

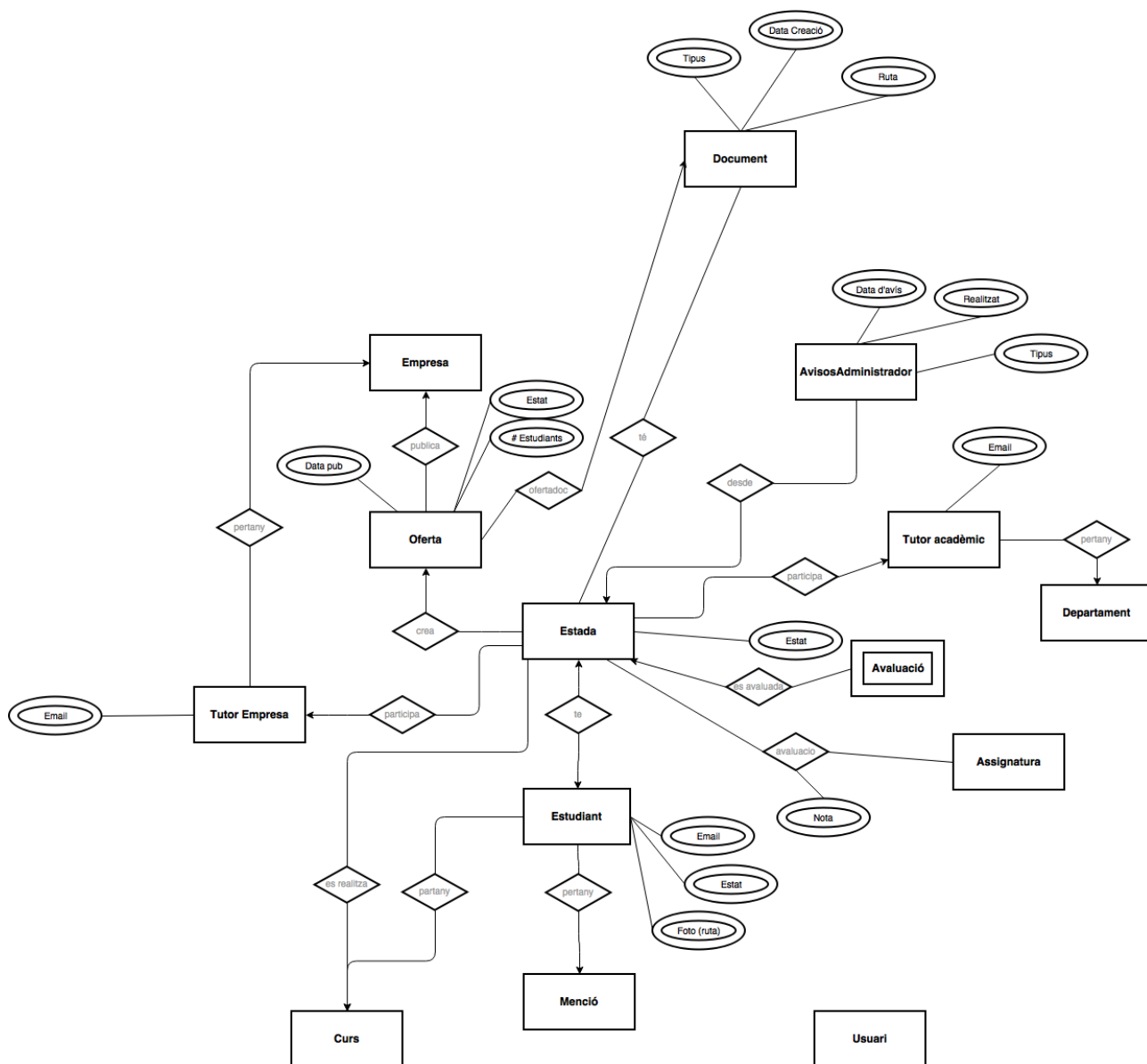
## REFERÈNCIES

- [1] Jordi Pons Aróztegui. Guia docent pràctiques externes. 2014-2015. URL <http://ddd.uab.cat/pub/procur/2014-15/g102766a2014-15iCAT.pdf>.
- [2] Joomla. Joomla main page. 2005 - 2015. URL <http://www.joomla.org>.
- [3] Wordpress. Wordpress main page. URL <https://es.wordpress.com/>.
- [4] Drupal. Drupal about. 2015. URL <https://www.drupal.org/about>.
- [5] InfoJobs. Infojobs main page. 2015. URL <https://www.infojobs.net/>.
- [6] Universidad Rey Don Juan Carlos. Manuals d'usuari, Març 2015. URL [http://www.urjc.es/practicas\\_externas/acceso.html](http://www.urjc.es/practicas_externas/acceso.html).
- [7] Mathias Weske. *Business Process Management*. 2012. doi: 10.1007/978-3-642-28616-2.
- [8] Jeff Sutherland. *Ken Schwaber Scrum guide*. 2013. URL <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf>.

- [9] Addison Wesley Kent Beck. *Extreme Programming Explained: Embrace Change*. 2000. ISBN 0-6201-61641.
- [10] English Prolinx Services Inc. *imatge, Agile software development Diagram*. 2013. URL <http://www.prolinxservices.com/agile.aspx>.
- [11] Melanie Pinola. *Cómo utilizar trello para organizar tu vida casi al completo*. 2015. URL <http://es.gizmodo.com/como-organizar-toda-tu-vida-utilizando-trello-1684529913>.
- [12] Laravel Dev Team. *Documentació del framework Laravel*. 2015. URL <http://laravel.com/docs/5.0>.
- [13] Google Inc. *Web fundamentals*. 2015. URL <https://developers.google.com/web/fundamentals/>.
- [14] Google Inc. *AngularJS landing Page*. 2015. URL <https://angularjs.org/>.
- [15] Angular Dev Team. *Documentació d'angular material*. 2015. URL <https://material.angularjs.org>.
- [16] Pete LePage. *Responsive web design*. 2014. URL <https://developers.google.com/web/fundamentals/layouts/rwd-fundamentals/?hl=en>.
- [17] Ethan Marcotte. *Responsive web design, a list apart*. Maig 2010. URL <http://alistapart.com/article/responsive-web-design>.
- [18] Scott Chacon and Ben Straub. *Pro Git (official git book)*. 2015. URL <https://progit.org/>.
- [19] Web Services Glossary W3c. 2004. URL <http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice>.
- [20] Fielding R. T. Taylor, R. N. *Principled design of the modern Web architecture*. 2000. doi: 10.1145/337180.337228.
- [21] The single page interface manifesto, 2014. URL [http://itsnat.sourceforge.net/php/spim/spi\\_manifesto\\_en.php](http://itsnat.sourceforge.net/php/spim/spi_manifesto_en.php).
- [22] James Shore. *Dependency injection demystified*, 2006. URL <http://www.jamesshore.com/Blog/Dependency-Injection-Demystified.html>.
- [23] Anthony Ferraral. *Request for Comments: Adding simple password hashing API*. URL [https://wiki.php.net/rfc/password\\_hash](https://wiki.php.net/rfc/password_hash).
- [24] Ecma International. *Standard ECMA-404 1st Edition, The JSON Data Interchange Format*. 2013. URL <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>.
- [25] Markus Gattol. *Time - Create, Read, Store, Manipulate*. 2015. URL <http://www.markus-gattol.name/ws/time.html#cron>.

## APÈNDIX

### A.1 Diagrama Entitat-Relació de la base de dades



## A.2 Diagrama lògic de la base de dades

